Providing Expert Advice in the Domain of Collaborative Scientific Inquiry[1]

Jozsef A. Toth, Daniel Suthers and Arlene Weiner

Learning Research and Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260 USA
INTERNET: jtoth+@pitt.edu, suthers+@pitt.edu, arlene+@pitt.edu
(412) 624-2189

Abstract

We describe an "expert coach" for students' collaborative information seeking and knowledge construction. This coach compares an expert's prespecification of the evidential relations among "snippets" of HTML-based text with the current state of a diagram constructed by students in the course of collaborative scientific inquiry. The expert advises the students by pointing out information that may challenge the students' views.

1 Introduction

This paper describes our current work in providing students with expert feedback as they engage in collaborative scientific inquiry. Our advice-giving subsystem, part of a larger system known as *Belvedere,* comprises two primary components: an argument "coach" [2], and an expert "coach" (see companion paper in these proceedings [7] for more details about the architecture). The students' task is to seek and integrate information about a scientific controversy. The system provides a facility for them to construct an "inquiry diagram" to represent the relations that they believe to hold between pieces of information. In the diagram, graphical shapes represent hypotheses, pieces of data, and principles. Arrows represent **FOR** and **AGAINST** relations. An **AND** link conjoins nodes that taken together have a relation to another node.

The argument coach provides advice about relationships among statements, but has nothing to say about the *contents* of these statements. Thus, it provides feedback about the "grammar" of scientific discourse, but knows nothing about the meaning of the diagrams. The argument coach analyzes the "inquiry diagram" as the students construct it. On request, the argument coach provides students with advice based on the structure of the diagram. The rules for the advice include syntactic advice on the conventions of the diagram and heuristics for scientific argument. For example, a scientific-heuristic rule is triggered when a hypothesis is supported by only one piece of data. The "one swallow does not a summer make" rule suggests that a good hypothesis explains more than one datum, and asks whether the student can find more data. Our system presently encodes twenty-three such structure-based rules, giving advice on a wide variety of situations involving: (A) statements about hypotheses and data, and (B) relationships about these statements using **FOR** and **AGAINST** links.

The domain-expert coach is a newer addition to the system and will be the primary focus of this paper. It was originally conceived as a *consistency-based* coach [5], but has since broadened in functionality, purpose and scope; hence, its new name, "expert coach." It is integrated with the argument coach in the new *Belvedere* system, and was delivered in January 1997 to five Department of Defense Education Administration (DODEA) schools in Europe.

The expert coach is assistive and Socratic. It assists students in their task by identifying information that might guide them in building a rich understanding. It is Socratic in preferentially providing information that is likely to create cognitive conflict and thus lead to deeper understanding. It does this by identifying paths through nodes in the expert's representation that differ from the corresponding paths in the representation the students are constructing.

[1] To appear in Proceedings of the Eight World Conference of the Artificial Intelligence in Education Society (AIED-97).

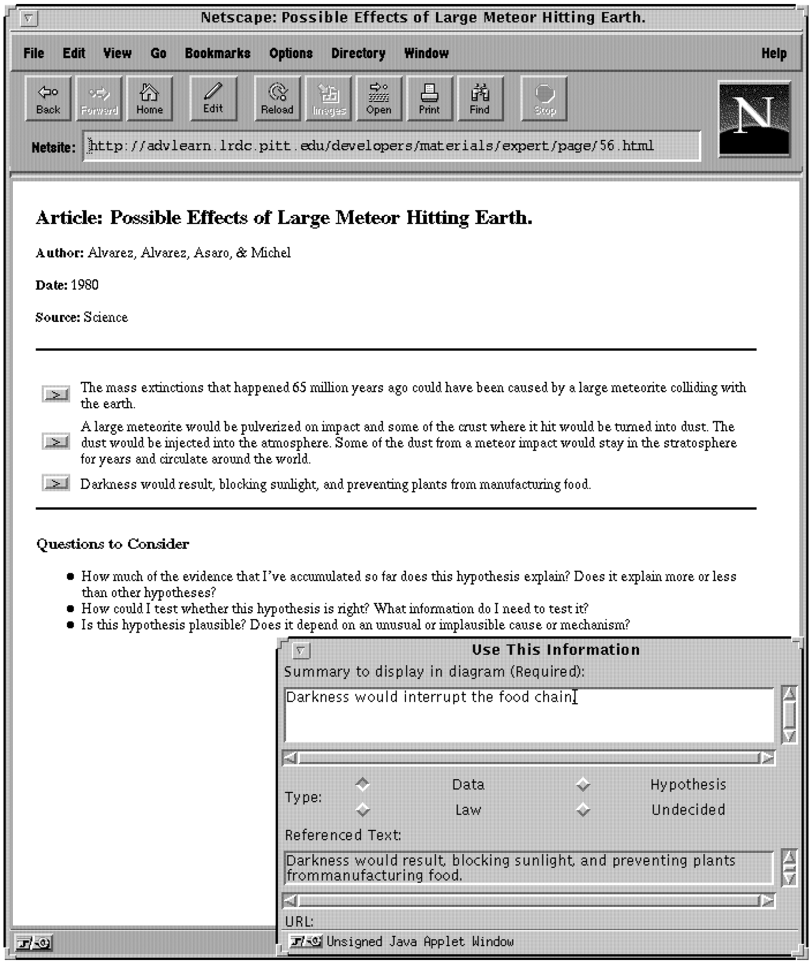

Figure 1: Client interface illustrating the selection of a "snippet" from a pre-composed HTML document with the Netscape browsing tool. This article has three selectable snippets about the meteorite impact theory of mass extinction.

2 Accessing Domain Information

Students using *Belvedere* access information from a prepared domain-relevant database of HTML-based documents browsed with the Netscape application. As they do, they can add new statements to the diagram and show their relation to other statements by using **FOR**, **AGAINST**, and **AND** links. Students have the choice of providing the contents of each statement either by typing in text or by selecting pre-coded HTML-based "snippets." A *snippet* is a unit of text that contains a statement or set of statements that can be used by students conducting a scientific inquiry. Typically, a snippet is a few sentences in length. To encourage deep processing, neither the contents of the snippet nor a predefined label is provided. Students must type in a short summary and identify the snippet as a **HYPOTHESIS, DATA**, or **PRINCIPLE**, before it is sent to an "In-Box" in the inquiry diagram.

We have presently encoded extensive domains of documents on two topics into HTML-based materials databases: *What caused mass extinctions?* and *What advice should be given to a person with a family history of a genetic disorder?* The Mass Extinctions database, for example, contains about one hundred snippets. Figure 1 illustrates how a student or teacher would select a preconfigured "snippet" from an HTML document while browsing the materials database using the Netscape browsing tool. Once such a snippet has been selected, a Java-based "applet" is activated that notifies the Belvedere client tool that the snippet has been selected. A visual icon known as the "In-Box" in the Client is highlighted, indicating that the material from this snippet can be relocated into the inquiry diagram (Figure 2). A list of such selections is maintained in this In-Box. Students also have the option to discard snippets present on the In-Box list. A dialogue box is again provided in which the student can check the snippet summary and modify its type, if necessary. When the

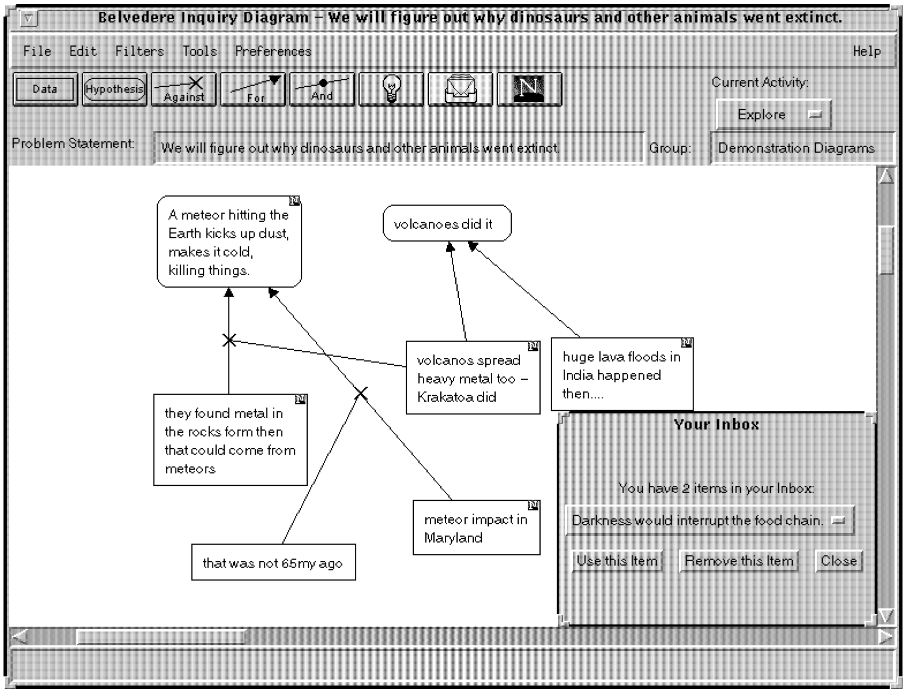

Figure 2: Client interface illustrating the selection of a snippet from the In-Box for inclusion in the inquiry diagram.

student retrieves the snippet from the In-Box, the student-written label (summary) appears in the appropriate hypothesis, data, or principle shape as defined by the student. In the diagram, a snippet is marked with a miniaturized version of the Netscape logo in the upper right-hand corner of the statement (Figure 2).

3 Mechanics of the Expert Coach

This most recent version of *Belvedere* integrates the LISP-based argument and expert coaches which run on a Sun Microsystems-based server, with Java-based client software running on Intel and Macintosh platforms. These machines are connected via a TCP/IP-based communications link (Figure 3). Currently, the relationship between the server-based coach and clients is one-to-many. The expert coach requires two knowledge bases, the student inquiry diagram and a corresponding expert inquiry diagram. Both kinds of diagrams are maintained in a Postgres relational database. At load time, these diagrams are read from the Postgres server into a LISP-based LOOM knowledge base and instantiated as LOOM objects. LOOM is a knowledge representation system developed and maintained at the Information Sciences Institute [1]. A set of expert diagrams is maintained by the expert (e.g., a teacher) and provides a canonical depiction of the teacher's mental model of the domain. The student and expert diagrams can consist of both snippets and "non-snippets"; i.e., text contents that are not predefined and so are not known to the expert coach.

The student cannot see the expert diagram during a session. The expert diagram is thought of as a "read-only" entity and is configured by the teacher before an inquiry session begins. The student diagram is dynamic; each time a change occurs in a student diagram, the change is noted by the expert coach and the LOOM knowledge base is updated with the new information.

As students construct an inquiry diagram, they may include snippets from the materials database. The expert coach is utilized only when a student assigns a relationship between two snippets with a **FOR**, **AGAINST**, or **AND** link. The expert coach can only provide advice about what it knows. Thus, if non-snippets are introduced into the student diagram the expert coach will be unaware of their existence, and it will have no advice about non-snippets created by the students. In that case, the argument coach still subjects the non-snippet objects to analysis and can respond. This interactivity between the coaches marks the genesis of a concept we call *advice arbitration* (Figure 3) in

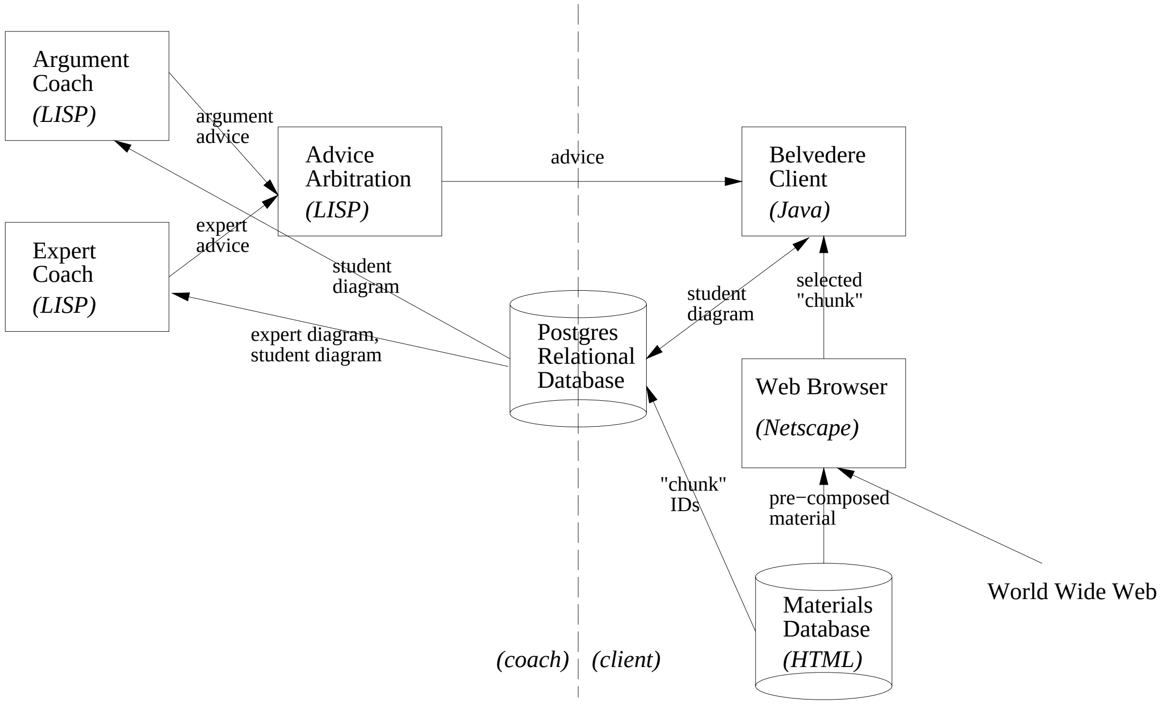

Figure 3: Interaction of student–client and coach components in *Belvedere* system.

which meta-rules manage the advice from two different coaches before the advice is passed on to the student. As advice is generated from each coach, it is maintained on a list which is then subject to a recursive multi-keyed "preference" sort. For instance, expert advice is provided first, multiple instances of the same advice are reduced to single instances, and argument advice is sorted according to type, ranging from "getting started" (e.g., the inquiry diagram is empty) to "advanced" (e.g., a swallow does not a summer make). In this fashion, we envision future arbitration schemes to manage several such knowledge sources.

The expert coach implies direction through the direction of the links in the diagrams. The *start* statement and *goal* statement, both snippets, are always related by a **FOR** or **AGAINST** link in the student diagram, and are utilized by the best-first search algorithm. For example if the data snippet **D1** is **FOR** or **AGAINST** the hypothesis **H1**, **D1** is the *start* snippet and **H1** is the *goal* snippet.

Following the earlier consistency algorithm from Paolucci et al. [5], the expert coach has been upgraded to utilize a best-first heuristic search to determine the optimal path from the *start* node to the *goal* node in the expert diagram using the cost function:

$f(n) = g(n) + h(n)$

where $g(n)$ is the distance of the path from the current node n in the graph back to the *start* node, and $h(n)$ is a heuristic estimate of the distance from the current node n to the *goal* [3]. The heuristic is articulated as follows: If the student has indicated a **FOR** link, all paths in the expert diagram which contain **AGAINST** links will be given shorter distances than paths with **FOR** links. Likewise, if a student has indicated an **AGAINST** link, all paths in the expert diagram which contain **FOR** links will be given shorter distances than paths with **AGAINST** links. In this fashion, as the best-first queue is sorted in non-decreasing order, the shorter paths will be sorted according to distance to the beginning of the queue and be favored at each new iteration of the search. The following rules also constrain the search of the expert coach:

1. The *start* and *goal* statements in the student diagram must be snippets and must also exist in the expert diagram.

2. The shortest path between the *start* and *goal* is always considered.

3. If the student has connected two snippets with a **FOR** link, and there is a path from the *start* to the *goal* in the expert inquiry diagram with an **AGAINST** link in it, provide feedback about

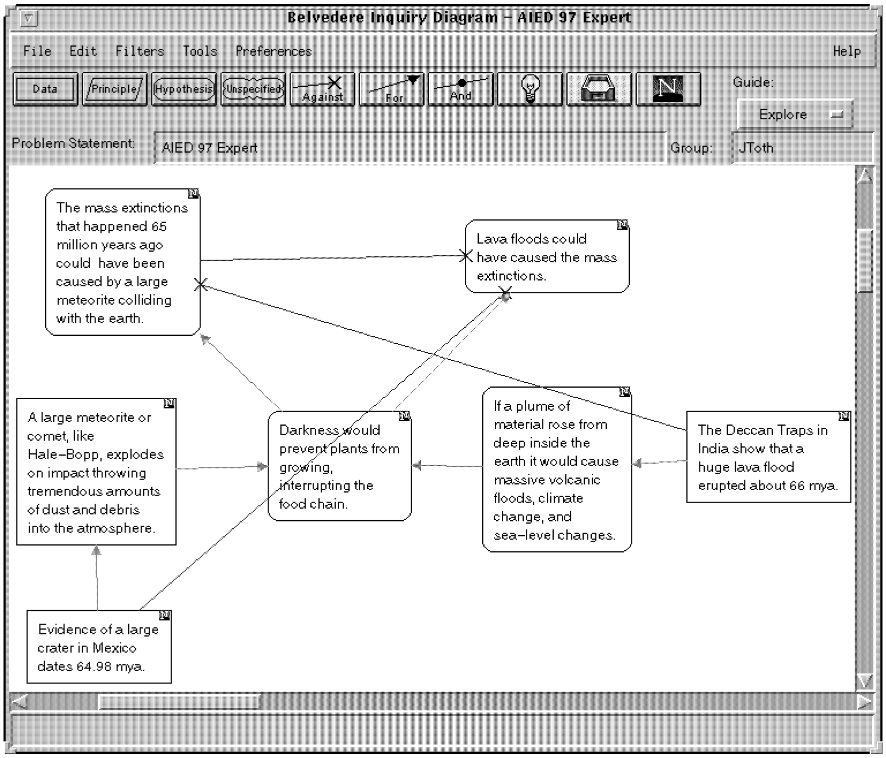

Figure 4: Example of an expert diagram illustrating two competing theories of mass extinction: meteorite impact and lava floods.

the contents of the two statements connected by the **AGAINST** link. The **AGAINST** link in the path closest to the *start* node is always preferred.

4. If the student has connected two snippets with an **AGAINST** link, and there is a path from the *start* to the *goal* in the expert inquiry diagram that consists entirely of **FOR** links, provide feedback about the contents of each statement in that path.

5. If either of the statements found in a path in the expert diagram is the *start* or the *goal* do not provide feedback about it.

6. If a student has indicated a **FOR** link between the *start* and *goal* and the expert diagram has a *direct* (i.e., no intervening links) **AGAINST** link between the *start* and *goal*, ask the student to consider the ramifications of an **AGAINST** link.

7. If a student has indicated an **AGAINST** link between the *start* and *goal* and the expert diagram has a *direct* (i.e., no intervening links) **FOR** link between the *start* and *goal*, ask the student to consider the ramifications of a **FOR** link.

Figures 4 and 5 illustrate sample expert and student diagrams, respectively. These diagrams are very simple in an attempt to illustrate the basic idea behind expert–student coaching. The expert diagram shows two competing hypotheses—meteorite impact and lava floods—relating to the topic of mass extinction. In Figure 5, the student has established a link, using a data-for-hypothesis relationship, suggesting that the Deccan traps in India provide support for the volcanic flood hypothesis. When queried, the expert coach examines the expert and student diagrams and finds the alternate hypothesis suggesting to the student that a meteorite could have also have caused the mass extinction. In the expert diagram, a path is found from the statement that reads “The Deccan Traps in India show that a huge lava flood erupted about 66 mya.”, which is derived from the same snippet as the data statement in the student diagram that reads “Deccan traps in India date 66mya.” This is the *start* snippet in the algorithm. The *goal* snippet is the hypothesis statement in both diagrams that reads

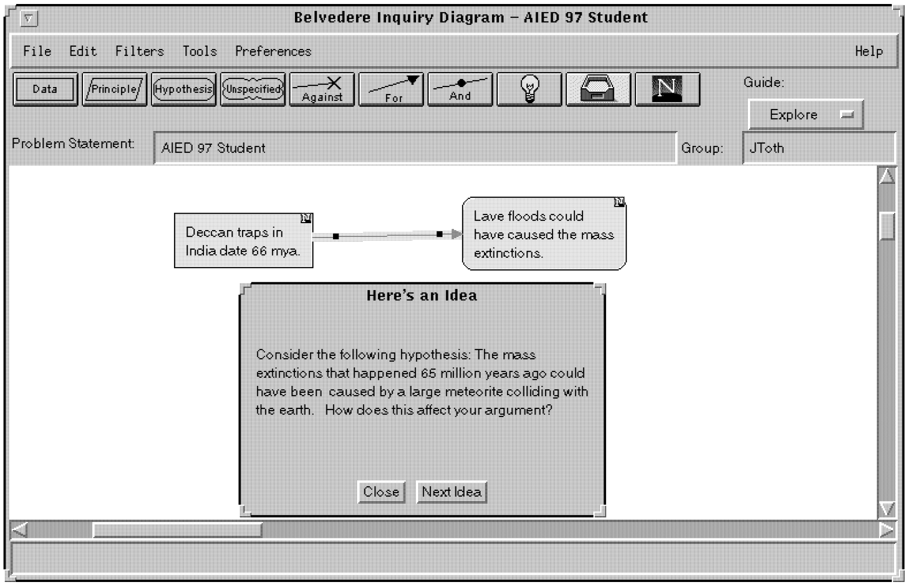

Figure 5: Example of expert advice given to student after establishing a data-for-hypothesis relationship. Advice is a pop-up dialogue box suggesting an alternative hypothesis found in expert diagram.

"Lava floods could have caused mass extinctions". In this case, the algorithm has found a path from the *start* statement to the *goal* statement containing a contradictory **AGAINST** link. In accord with the heuristics enumerated above, the reference text from the HTML document in the snippet adjacent to the **AGAINST** link becomes the advice that is provided to the student, suggesting the alternate hypothesis, meteorite impact; something the student may not have yet considered. Note that this snippet's origin is in the HTML document illustrated in Figure 1 (the first of three snippets in that illustration).

4 Future Directions

We are presently porting the LOOM LISP-based coaching code to the Java programming language. The bulk of the student software is already written in Java and we would like to achieve the platform-independence that Java affords with as many components of the system as we can. The LOOM run-time image consumes tens of megabytes of memory and is platform-dependent. We have tentatively selected the "Java Expert System Shell" (JESS), which is a rule-based inference engine utilizing the Rete pattern-matching algorithm [4]. Jess is written entirely in Java. At the time of this writing, twenty-one of the twenty-three LOOM-based rules for the argument coach have been re-implemented in JESS. What remains is to interface the client software with the JESS coaching system and to replicate the preference sort in Java. We can then proceed to re-implement the expert coach and add some extensions which will be described in a moment.

Although we have selected an appropriate level of representation, the snippet, to allow the student to access domain-relevant material, we are considering the pedagogical value of both a finer and a coarser grain size. A finer grain would reduce ambiguity and increase the accuracy of feedback. On the other hand, a coarser grain, i.e., at the level of a normal paragraph, or of a typical Web document, would enable quicker authoring of the Web-based materials described earlier. Currently the expert's specification of the relations is a major bottleneck for complex domains. The model of coaching with a larger grain size would be an "FYI" coach, which would function like a research librarian forwarding new information to those likely to be interested in it. It would still be possible to specify **FOR** and **AGAINST** relations in a general sense, just as a paper can give evidence for or against a particular view. However, coarse-grained representation has obvious limitations. For example, it is important for students to learn that one can often extract evidence *for* a view from a context that is generally unfavorable. Indeed, scientific papers are obliged to take note of divergent views and limitations. We are also considering exposing the student to sub-graphs of the expert diagram. We are exploring models of learning and cognitive/perceptual mapping for the novice and expert, regarding the information realized in the diagrams the web-based materials (e.g., [6]).

We are also working on the idea of extending and enhancing the interaction between the argument and expert coaches. Presently, we only evaluate the status of **FOR** and **AGAINST** relations between adjoining nodes in the search of a path from the *start* node to a *goal* node. We would also like to imbue the semantics of the best-first search to include higher-order structures involving more than one relation and more than two statements. This would not be unlike the pattern-matching strategies that are presently employed in the argument coach, but which would have to be integrated with the best-first algorithm. At the time of this writing, we have prototyped a few basic patterns against which the expert coach compares, for example, a basic data-for-hypothesis structure in the student diagram, with the same structure (using the same snippets, etc.) in the expert diagram, and then notifies the student that other data or hypotheses exist that either support or refute the hypothesis or data in the basic structure.

We also seek better integration between the *Belvedere* client interface and the Netscape browser. When advice related to a snippet is provided to the student, we would like to have the appropriate HTML-based reference document automatically appear in the Netscape browser. A further research direction is to integrate acquisition of pre-existing HTML documents with authoring tools that allow an expert to specify their relations.

5 Acknowledgements

Thanks to Alan Lesgold, Dan Jones, Kim Harrigal, and Massimo Paolucci. This work was funded by the Advanced Research Projects Agency Computer Aided Education and Training Initiative, under the title "Collaboration, Apprenticeship, and Critical Discussion: Groupware for Learning," Contract N66001-95-C-8621.

References

[1] R Bates and R MacGregor. The loom knowledge representation language. ADA 183415 RS-87-188, Information Sciences Institute, University of Southern California, Marina del Ray, CA, 1987.

[2] V Cavalli-Sforza, A Weiner, and A Lesgold. Software support for students engaging in scientific activity and scientific controversy. *Science Education*, 78(6), 1994.

[3] P R Cohen and E A Feigenbaum. *The Handbook of Artificial Intelligence*, volume 1. Addison-Wesley, New York, 1989.

[4] E. Friedman-Hill. *Jess, the Java Expert System Shell*. Sandia National Laboratories, Livermore, CA, 1997. http://herzberg.ca.sandia.gov/.

[5] M Paolucci, D Suthers, and A Weiner. Automated advice-giving strategies for scientific inquiry. In C Frasson, G Gauthier, and A Lesgold, editors, *Intelligent Tutoring Systems, Third International Conference, ITS'96 Montreal, Canada, June 1996*, Lecture Notes in Computer Science, pages 372–381. Springer, New York, 1996.

[6] M Petre and T R G Green. Learning to read graphics: Some evidence that 'seeing' an information display is an acquired skill. *Journal of Visual Languages and Computing*, 4:55–70, 1993.

[7] D Suthers and D Jones. An architecture for intelligent collaborative educational systems. In *Proceedings of the Eight World Conference of the Artificial Intelligence in Education Society (AIED-97)*, Kobe, Japan, August 19–22 1997.